

Replace the paragraph starting on line 9 through line 12 on page 7 of the Brief Description of the Drawings section with the following two paragraphs:

FIG. 3 illustrates a configuration of an analytical virtual machine running PE, NE or LE Windows WINDOWS executable code (code that produces rectangular graphics containing a graphical user interface).

FIG. 4 shows a memory map onf an analytical virtual machine running a binary (COM or SYS) executable and running a Visual Basic VISUAL BASIC (VB) executable (a software program designed to facilitate the development of systems with a graphical user interface).

[The remainder of this page has been intentionally left blank.]

Replace the paragraph starting on line 16 of page 15 through line 25 of page 16 of the Detailed Description of the Preferred Embodiments section with the following paragraph:

The IP register points to the entry point in the virtual memory. At this point in the virtual execution process, the virtual CPU starts to fill the 12-byte prefetch queue. The bytes are decoded according to Intel's Pentium PENTIUM (high-performance microprocessor) instruction set reference, whereby the first byte in this 12-byte queue determines the function of the instruction word. The instruction word comprises an operation, an interrupt call or an operating system API call. Of course, in implementations for other processors, other instruction set references will be appropriate and the specific details will vary. It will be apparent from this discussion how to implement the analytical virtual machine for different processors and different operating system implementations.

[The remainder of this page has been intentionally left blank.]

Replace the paragraph starting on line 1 through line 12 on page 19 of the Detailed Description of the Preferred Embodiments section with the following paragraph:

The system performs the sequence of events for each instruction line, and continues at each entry point in the code until all entry points in the entry point table have been processed. The AVM then returns the behavior pattern, together with the sequencer structure, to the calling program and the AVM is terminated. The process flow of the FIG. 2 instance of the AVM is further illustrated in FIG. 7, which shows the processing flow within the current AVM implementation after the AVM has been initialized to interpret and extract the behavior pattern of a high level language (HLL) program, such as ~~VB Script~~ VB SCRIPT code (scripting language based on Visual Basic, a software program designed to facilitate the development of systems with a graphical user interface). The language definition key-word list and rules define the language interpreted by the processing core. In FIG. 7, a keyword list is shown that is a cross between VBA and VBS, but the processing core is not limited to those languages.

[The remainder of this page has been intentionally left blank.]

Replace the paragraph starting on line 17 of page 20 through line 23 of page 21 of the Detailed Description of the Preferred Embodiments section with the following paragraph:

In a present implementation, an analytical virtual machine in accordance with the present invention has three operating modes, a high level mode, a protected mode and a real mode. The real mode corresponds to an operating system such as DOS and the corresponding instance of the AVM is illustrated in FIGS. 1 and 6. The high level mode might be used to analyze a program in a high level language such as Perl PERL (Practical Extraction and Report Language), Visual Basic, or a scripting language. A high level mode of the AVM is illustrated in FIGS. 2 and 7.

[The remainder of this page has been intentionally left blank.]